Fu G. FCA based ontology development for data integration. *Information Processing & Management* 2016, DOI: 10.1016/j.ipm.2016.02.003

Information Processing and Management 000 (2016) 1–18

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

FCA based ontology development for data integration

Gaihua Fu*

School of Civil Engineering and Geosciences, Newcastle University, Newcastle upon Tyne, UK

ARTICLE INFO

Article history:
Received 12 May 2015
Revised 5 November 2015
Accepted 22 February 2016
Available online xxx

Keywords:
Ontology development
Formal concept analysis
Data integration
Information sharing

ABSTRACT

Data is a valuable asset to our society. Effective use of data can enhance productivity of business and create economic benefit to customers. However with data growing at unprecedented rates, organisations are struggling to take full advantage of available data. One main reason for this is that data is usually originated from disparate sources. This can result in data heterogeneity, and prevent data from being digested easily. Among other techniques developed, ontology based approaches is one promising method for overcoming heterogeneity and improving data interoperability. This paper contributes a formal and semi-automated approach for ontology development based on Formal Concept Analysis (FCA), with the aim to integrate data that exhibits implicit and ambiguous information. A case study has been carried out on several non-trivial industrial datasets, and our experimental results demonstrate that proposed method offers an effective mechanism that enables organisations to interrogate and curate heterogeneous data, and to create the knowledge that meets the need of business.

1. Introduction

Business productivity and competitiveness are increasingly being driven by the effective access and use of data. Data provides a mine of information that can help us spot undiscovered patterns of business importance and to create the knowledge that will be needed to tackle the challenges of future. However with data becoming available and growing at unprecedented rates, organisations struggle to take full advantage of valuable data. One main reason for this is that data is usually created and maintained by a range of organisations. This results in mismatch between datasets, i.e., datasets differ from one organisation to another not only in *what* is encoded but also in *how* it is encoded.

In order for organisations to use and digest heterogeneous data and uncover the untold business patterns, there is a growing interest to develop techniques that investigate complex data phenomena and facilitate better data interoperability (Doan, Halevy, & Ives, 2012; Doan, Noy, & Halevy, 2004; Duckham & Worboys, 2005; Huang, Lin, & Chan, 2012; Jiang, Zhang, Tang, & Nie, 2015; Lenzerini, 2002). Among various techniques developed, ontology research is one discipline that can deal with data heterogeneity and improve data sharing (Kalfoglou & Schorlemmer, 2003; Mate et al., 2015; Noy, 2004). Ontology-based integration systems are usually characterised by a *global* ontology which represents a reconciled, integrated view of the underlying data sources. Systems taking this approach usually provide users with a uniform interface—all queries made to source data are expressed in terms of a global ontology, as are the query results. This frees the user from the need to understand each individual data source. Unfortunately, in many domains one faces the problems of either having no

* Corresponding author. Tel: +441912086822.
E-mail address: Gaihua.fu@ncl.ac.uk, fugaihua@hotmail.co.uk

http://dx.doi.org/10.1016/j.ipm.2016.02.003
0306-4573/

established ontology that can be readily employed in the integration work, or existing ontologies do not fit for the purpose (e.g., not consisting of knowledge that sufficiently captures the semantics of the information under investigation).

In this paper we contribute a formal and semi-automated approach for ontology development. Rather than starting from scratch, we build an ontology by effective discovering and use of the knowledge that is buried in the datasets to be integrated. The method is based on Formal Concept Analysis (FCA) (Ganter & Wille, 1999; Ganter, Stumme, & Wille, 2005), which is a mathematical approach for data analysis. FCA supports ontology development by abstracting conceptual structures from attribute-based object descriptions, and it enables considerable ontology development activities automated.

Our research extends classical FCA theory to support ontology development for integrating datasets that exhibit *implicit* and *ambiguous* information. Implicit information is caused by the fact that some organisations tend to take some domain knowledge as granted, and do not explicitly specify it in their design documents or datasets. This can lead to an ontology that is ill-formed, and does not correctly capture critical concepts and the semantics of the domain. Ambiguous information is due to the fact that organisations differ from each other in culture, conventions and requirements in system development, hence they may vary in how they choose to represent a business object, and at what levels of granularity such information is encoded. This causes inconsistencies between the datasets of different organisations.

We consider that overcoming this implicity and ambiguity is an important step in ontology development. The work reported here is a follow on research of Beck et al. (2013), Fu and Cohn (2008a) and Fu and Cohn (2008b). In this paper we report further technical advances we have made. To restore implicit information, we introduce a rule based method. We discuss how rules are derived and deployed for recovering implicit information. To resolve disambiguate information, we define a set of primitive operations to deal with simple matches in data alignment. These operations are then composed to deal with more complicated matches. Finally, we report on our experiments that are carried out to construct an ontology for integrating non-trivial datasets from several UK water companies. We measure the quality of the developed ontology by utilising the metrics of classical information theory and also in terms of its *fitness* to the application domain. Our experimental results demonstrate that techniques described in this paper provide an effective mechanism for reconciling and harmonising heterogeneous data from disparate sources, and they support development of ontologies that better *fit* and *respect* the underlying knowledge structures of domains.

The remaining part of the paper is organised as follows. Section 2 reviews related research. Section 3 recalls relevant notions of FCA and briefs our framework for ontology development. Sections 4 and 5 present techniques that deal with implicit and ambiguous information. Section 6 discusses how to derive an ontology by using results generated from Sections 4 and 5. Section 7 reports our experimental results. Section 8 concludes the paper and suggests future research.

2. Related research

Several areas of research are interesting to this work. Firstly, integration techniques investigated in database and information integration are quite relevant. Various topics have been studied by these communities and the ones that are the most interesting here are *mapping discovery* and *schema integration*, and techniques have been developed to support these (Bahga & Madisetti, 2015; Do & Rahm, 2002; Doan et al., 2004; Lenzerini, 2002; Liu & Zhang, 2014; Madhavan & Halevy, 2003; Pedersen, Pedersen, & Riis, 2013; Rahm & Bernstein, 2001). *Mapping discovery* takes two or more database schemas as input and produces a mapping between elements of the input schemas that correspond semantically to each other. Many of the early as well as current mapping solutions employ hand-crafted rules or heuristics to match schemas (Madhavan, Bernstein, & Rahm, 2001; Rahm & Bernstein, 2001). Examples of such heuristics include linguistic matching of schema element names, detecting similarity of structures of schema elements, and considering the patterns in relationships of the schema elements. Techniques have also been proposed to use learning based methods (Doan, Domingos, & Halevy, 2001; Neumann, Ho, Tian, Haas, & Meggido, 2002).

Schema integration constructs a global schema based on the inter-schema relationships produced in mapping discovery. Each mapping element is analysed to decide which representation of related elements should be included in the global schema. When a mapping describes the corresponding schema elements as identical, their integration is straightforward—simply includes one of schema elements into the global schema. More frequently, the corresponding schema elements are not the same but are mutually related by some semantic properties, and schema merging is performed manually or semi-automatically with the assistance of domain engineers to guide the designers in their resolution.

Ontology research is another discipline that deals with data integration. A common definition of an ontology is that it is a formal, explicit specification of a domain of discourse (Gruber, 1993). As it provides a shared understanding and explicit specification of a domain, an ontology is considered to have a key role to play in data integration (Bakhtouchi, Bellatreche, & Ait-Ameur, 2011; Bian, Zhang, & Peng, 2011; Noy, 2004; Uschold & Grüninger, 2004; Yu et al., 2012). Unfortunately, for many domains one faces the need to develop ontologies from scratch (as there is no existing ontology that can be used readily), and a growing number of methods have been proposed in recent years to address the issues of ontology design and development. Most methods are based on the traditional knowledge engineering approach (Brockmans et al., 2006; Pinto & Martins, 2004; Sure, Tempich, & Vrandecic, 2006). These methods usually start with defining the domain and scope of ontologies. This is followed by a data acquisition process: important concepts are collected; a concept hierarchy is derived, and properties and semantic constraints are attached to concepts.

As developing ontologies from scratch is an expensive process to perform, there has been increasing interest in reusing or merging existing ontologies (or other knowledge structures such as thesauri) that are developed independently in different

Please cite this article as: G. Fu, FCA based ontology development for data integration, Information Processing and Management (2016), http://dx.doi.org/10.1016/j.ipm.2016.02.003

applications (Duong, Truong, & Nguyen, 2012; Truong & Nguyen, 2012; Xie, Liu, & Guan, 2011; Yang, 2011). Central to these studies is research on ontology mapping and ontology integration. Approaches to ontology mapping are similar to ones for matching database schemas and other structured data, and they use lexical and structural components of definitions to find correspondences. However, as an ontology captures richer data semantics than traditional database schemas, the methods for finding mappings tend to exploit these extra data semantics (Kalfoglou & Schorlemmer, 2003; Nguyen, 2007; Rodriguez & Egenhofer, 2003; Truong & Nguyen, 2012). For example, in Noy and Musen (2000) a tool has been developed to use linguistic similarity matches between concepts for initiating mappings, and then use the underlying ontological structures (classes, slots, facets) to suggest a set of heuristics for identifying further matches between the ontologies. In Duong and Jo (2012), a method has been proposed to mapping ontological concepts using propagating Priorly Matchable Concepts. The method exploits information such as concept types, relations and constraints to provide suggestions for possible concept matches. The method guilds on how to priorly check the similarity between concepts and it reduces computational complexity by avoiding checking similarity among unmatchable concepts. In Nguyen (2006), an approach has been proposed to resolve three levels of ontology conflicts: instant level, concept level and relation level, using consensus method. The techniques developed in Doan, Madhavan, Domingos, & Halevy (2003) and Spohr, Hollink, and Cimiano (2011) employs learning based techniques to find ontology mappings. They exploit information in data instances and taxonomic structure of ontologies, and then uses a probabilistic model to combine results of different learners.

Based on the inter-ontology mappings derived in mapping discovery, a merging process integrates the source ontologies and generates a global ontology. However, deriving a meaningful ontology is a hard problem even with the ground set of inter-ontology mappings provided, and most methods that support the merging process are performed in an interactive manner with the assistance of human users, as is done in database and information integration research.

Another branch of research studies ontology development and integration with formal methods. Of particular interest here is research based on Formal Concept Analysis (FCA) (Ganter & Wille, 1999; Ganter et al., 2005; Wille, 1982). FCA is a formal method for concept classification and conceptual structure derivation. FCA related tools enable considerable knowledge processing activities to be automated, particularly concept generation and hierarchy derivation. As a result, FCA has been attracting great interest to support systematic, semi-automated development and integration of ontologies (Bai & Zhou, 2011; Formica, 2006; He & Wang, 2011; Nanda, Simpson, Kumara, & Shooter, 2006; Xia, 2013). For example, in Rouane, Valtchev, Sahraoui, & Huchard (2004) ontological hierarchy merging is studied in the framework of FCA by taking into account of both taxonomic and other semantic relationships of ontologies. A method FCA-MERGE has been developed in Stumme and Maedche (2001) to use FCA to support ontology integration. FCA_MERGE takes as input the two ontologies and a set of natural language documents, and computes a concept lattice from two source ontologies using FCA techniques. The concept lattice is then exploited by domain experts to derive a merged ontology. In Zhao, Wang, and Halang (2006) a similarity method has been introduced to map ontology concepts basing on Rough Set and Formal Concept Analysis theory. The idea is to construct from two source ontologies a concept lattice with FCA and similarity measure of two concepts are then computed using Rough Set theory. In Chen, Bau, and Yeh (2011) authors proposed a method that combines WordNet and Fuzzy Formal Concept Analysis techniques for merging ontologies. WordNet is firstly used to align concepts from a source ontology to concepts in a base ontology, and the remaining unmapped concepts are then aligned to the base ontology using a similarity measure based on fuzzy FCA.

Our approach is in line with FCA based research. Yet it differs from previous studies in several aspects. Firstly, while most research focusing on similarity measure of ontology concepts, we contribute an integrated framework that offers a structural and systematic description of ontology merging process. Secondly, with FCA as backbone we investigate how to resolve implicit and ambiguous information. Previous research is either implicit on how these problems are resolved, or only address particular types of these problems. For example, in Rouane et al. (2004) there is an interesting discussion on attribute conflicts, but the authors do not address in detail how these problems are resolved. Thirdly, while most previous research considers one to one mapping between concepts, our method is able to deal with more complicated issues, i.e., an ontology concept may have multi-mappings from another ontology, which has not been investigated sufficiently in literature. Finally, we applied the proposed techniques to non-trivial industrial datasets, and examined how effectively the proposed method can help with improving data interoperability. This has rarely been reported in other FCA-based works.

3. FCA terminologies and an FCA based framework for ontology development

In this section, we introduce the basic concepts of FCA and brief our framework for ontology development. We will use data and examples from water infrastructure domain to present techniques developed in this research.

FCA theory was developed in Wille (1982) and a typical task that FCA can perform is data analysis, making the conceptual structure of the data visible and accessible (Ganter & Wille, 1999; Ganter et al., 2005). Central to FCA is the notion of *formal context*, which is defined as a triple $K{:=}\langle G, M, I\rangle$, where G is a set of objects, M is a set of attributes, and $I{\subseteq}GXM$ is a binary relation between G and M. A relation $\langle g, m\rangle \in I$ is read as "object g has the attribute m". A formal context can be depicted by a cross table as shown in Fig. 1(a), where the elements on the left side are objects; the elements at the top are attributes; and the relations between them are represented by the crosses.

A formal concept of a context $K{:=}\ \langle G, M, I\rangle$ is defined as pair (A, B), where $A{\subseteq}G$, $B{\subseteq}M$, $A' = B$ and $B'{=}A$. A' is the set of attributes common to all the objects in A and B' is the set of objects having the attributes in B. The *extent* of the concept (A, B) is A and its *intent* is B. The formal concepts of a context are ordered by the sub- and super-concept relations. The set of

Please cite this article as: G. Fu, FCA based ontology development for data integration, Information Processing and Management (2016), http://dx.doi.org/10.1016/j.ipm.2016.02.003

a

	A1	A2	A3	A4
O1	X	X		X
O2		X	X	X
O3	X	X	X	
O4	X		X	

b

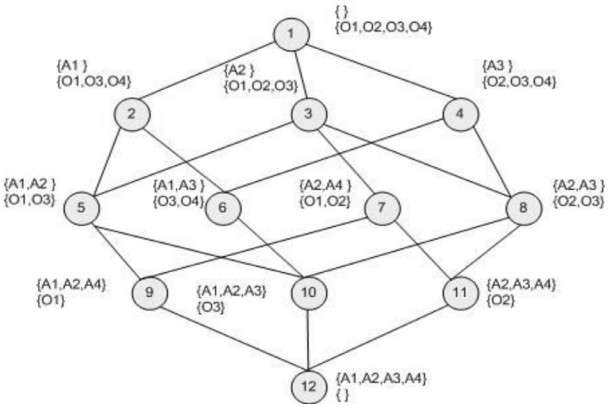

Fig. 1. (a) A formal context. (b) The concept lattice for the context in (a).

a

	what	how
pipeType1	foul	gravity
pipeType2	sludge	pressure

b

	foul	sludge	gravity	pressure
pipetype1	X		X	
pipetype2		X		X

Fig. 2. (a) A many valued context. (b) A one value context after conceptual scaling.

all formal concepts ordered by sub- and super-concept relations forms a concept lattice. Fig. 1(b) shows the concept lattice for the context in Fig. 1(a), where a node represents a concept labelled with its intensional and extensional description. The links represent the sub- and super-concept relations.

The formal contexts introduced above are not the ones that occur most frequently in applications of FCA. Most often data is encoded in *many valued contexts*. A many valued context $K:= \langle G, M, W, I\rangle$ consists of a set of objects G, a set of attributes M, a set of attribute values W, and a set of ternary relations $I \subseteq G \times M \times W$. A relation $<g, m, w> \in I$ is read as "object g has the attribute m and its value is w". Fig. 2(a) shows a many valued context which lists different water pipes having different attribute values. In order for FCA theory to be applied to a many valued context, it needs to be unfolded into a one valued context through conceptual scaling (Ganter & Wille, 1999). Fig. 2(b) shows the one valued context for the many valued context in Fig. 2(a) after conceptual scaling.

As the extent and intent of a concept overlaps with those of its super- and sub-concepts, redundancy exists in a concept lattice. To prevent this, *reduced labelling* is introduced. A lattice with reduced labelling is obtained by replacing each concept *(A,B)* with *(N(A),N(B))*, where *N(A)* contains the non-redundant elements in *A*, and *N(B)* contains the non-redundant elements in *B*. An object *o* will appear in *N(A)* if the corresponding concept is the greatest lower bound of all concepts containing *o*. An attribute *a* will appear in *N*(B) if the corresponding concept is the least upper bound of all concepts containing *a*. Fig. 3(a) shows the lattice derived from the one in Fig. 1(b) with reduced labelling. Furthermore we can eliminate in a lattice the concepts which do not possess their own attributes or objects. This leads to a structure called a *Galois Sub Hierarchy (GSH)*. A GSH only consists of so called *attribute concepts* and *object concepts*. An object concept represents the smallest concept with this object in its extension, and an attribute concept represents the largest concept with this attribute in its intension. The GSH of the lattice in Fig. 3(a) is depicted in Fig. 3(b), where concepts 1, 5, 8 and 12 are removed due to the empty *N*(*A*) and *N*(*B*). Concept 2 is an attribute concept and concept 9 is an object concept.

With the FCA theory as the backbone, we have developed a framework to support ontology development. The framework essentially consists of three components: Context Formation, Context Composition and Ontology Derivation, as illustrated in Fig. 4. To generate an integrated ontology for two datasets, Context Formation takes the datasets as inputs and generates a one valued context for each of them. The generated contexts are then fed to Context Composition to produce an integrated GSH. Ontology Derivation takes the GSH generated in Context Composition and generates an integrated ontology as well as concept mappings between two datasets. We will describe Context Formation in Section 4, and elaborate on Context Composition and Ontology Derivation in Sections 5 and 6.

4. Context formation

Fig. 5 shows the components of Context formation. Given a dataset, Data Acquisition derives concepts encoded in the dataset as well as their attribute definitions, and the result is a many valued context for the dataset. The component looks at sources where various feature types (concepts) and their definitions can be extracted. The most common sources here are

Please cite this article as: G. Fu, FCA based ontology development for data integration, Information Processing and Management (2016), http://dx.doi.org/10.1016/j.ipm.2016.02.003

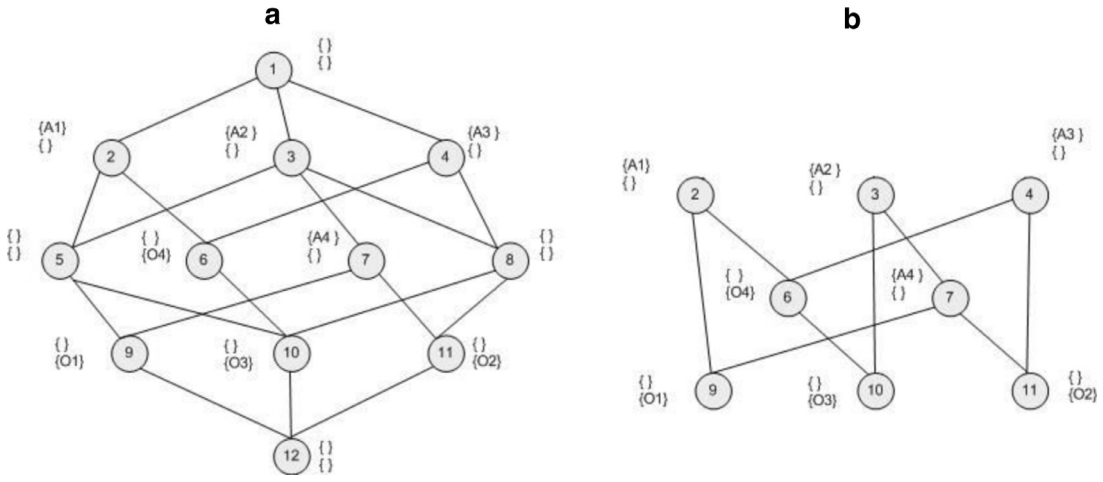

Fig. 3. (a) A lattice with reduced labelling. (b) A Galois sub hierarchy.

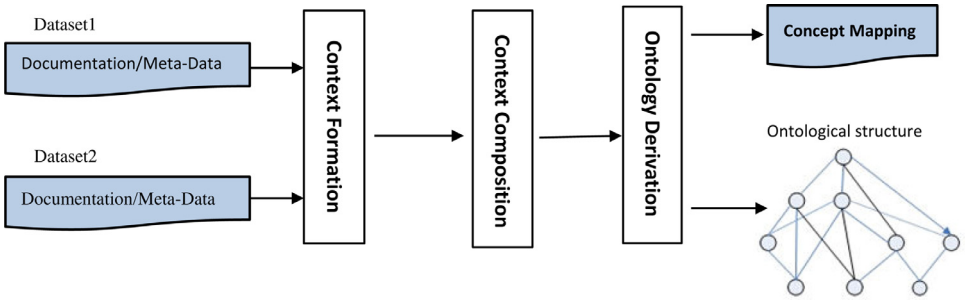

Fig. 4. An FCA based framework for ontology development.

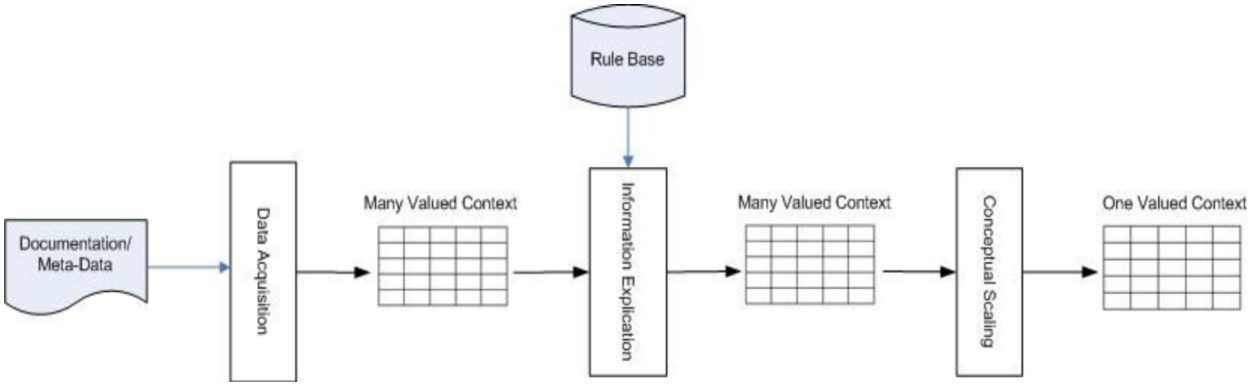

Fig. 5. Context formation.

text/web documents created by system designers/developers for specifying system requirements and design. Other important sources are conceptual/logical data models of the concerned dataset. The generated context is then fed to the Information Explication component to restore implicit information. The component Conceptual Scaling transforms a many valued context into a one valued context, in order for classic FCA techniques to be applicable.

4.1. Implicit information

The main challenge here is to deal with implicit information. Implicit information is caused by several factors. As an example in water infrastructure domain, when defining a feature type, organisations tend to explicitly state specific properties, but leave common ones unarticulated in their design documents. For instance, a sewer pipe is characterised by how it conveys sewage: either by *gravity* or by *pressure*, with the gravity distribution employed more often than the pressurised form.

Please cite this article as: G. Fu, FCA based ontology development for data integration, Information Processing and Management (2016), http://dx.doi.org/10.1016/j.ipm.2016.02.003

Table 1
A many valued context table.

	Size	What	How	Location
pipeType1	Main		Pressurised	
pipeType2	Main			Aboveground
pipeType3	Main	Sludge		
pipeType4	Main			

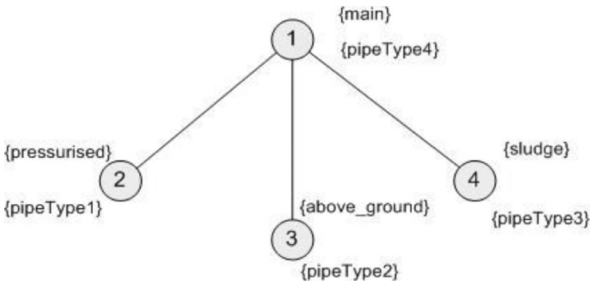

Fig. 6. The GSH for the formal context in Table 1.

Most water companies explicitly specify the pressurised characteristic of a sewer pipe, but not the gravity one. Furthermore, many organisations take some domain knowledge as granted, and do not encode it explicitly. For example, a sludge sewer is usually *pressurised* rather than *gravity*. As this is well understood in the domain, many water companies choose not to encode this information explicitly. Table 1 shows a portion of a many valued context that is generated for a sewerage dataset, where many blank cells exist due to implicit or unarticulated domain knowledge.

The main consequence of this is that it can lead to an ontology that is ill-formed, and does not correctly capture critical concepts and semantics of the domain. Fig. 6 shows the GSH for the context in Table 1. Due to implicit information, many important concepts, such as *gravity sewer* and *underground sewer*, are missing from the hierarchy and therefore from the resultant ontology. Furthermore, different organisations may choose what not to articulate in their datasets. We believe this hidden knowledge is one of main reasons that hinder data compatibility or interoperability across organisations.

4.2. Rule elicitation

We classify implicit information into two groups: attribute-specific and object-specific. Attribute-specific implicit information is concerned with a particular attribute, and is applicable to all objects having that attribute. Object-specific implicit information is concerned with an attribute of particular objects only. An example of former is with the *how* attribute in Table 1. The unarticulated domain knowledge here is that *a sewer pipe carries sewage by gravity if not explicitly specified,* and this applies to all sewerage pipes having *how* attribute. An example of object-specific implicit information is with the *how* attribute of *pipeType3.* The implicit information here is that *if a pipe carries sludge sewage, by default it carries it by pressure.* This is relevant to the *how* attribute, but applies to *pipeType3* only (pipes that carry sludge) and therefore is classified as object-specific implicit information.

We use a rule based approach to recover implicit information. As implicit information is largely unarticulated domain knowledge, we need to work closely with domain experts to acquire these rules. We have two types of rules, attribute rules dealing with attribute-specific implicit information, and object rules dealing with object-specific implicit information.

To elicit attribute rules, we iterate each attribute. An attribute has implicit information if it has *missing* values for some objects. Each attribute with implicit information in a context table incurs a rule. Involvement of domain experts is required at this point to generate such a rule. For example, Rule 1 in Fig. 7 is collected for the *how* attribute in Table 1.

To elicit object rules, we iterate each object in the context, and examine each of its attributes that do not have a value. If an attribute has implicit information which cannot be recovered with an attribute rule, an object rule is elicited to recover implicit information with the help of domain experts. For example, for object *pipeType3,* the attribute *how* has implicit

1) a sewerage pipe usually carries waste water unless it is specified otherwise;
2) if a sewerage pipe is not explicitly specified as a pressurised pipe, then it is a gravity sewer;
3) if a sewerage pipe is not explicitly specified as an above ground pipe, then it is an underground pipe;
4) a sludge sewerage pipe is a pressurised pipe unless it is specified otherwise;

Fig. 7. Rules for restoring implicit information for the context table in Table 1.

Please cite this article as: G. Fu, FCA based ontology development for data integration, Information Processing and Management (2016), http://dx.doi.org/10.1016/j.ipm.2016.02.003

information. As the implicit information for *how* in this case is *pressurised,* it cannot be recovered with Rule 1 discussed above. An object rule, Rule 4 in Fig. 7, is acquired in this case for *pipeType3*.

Fig. 7 shows a set of rules elicited for the context table in Table 1, where Rule 1, 2 and 3 are attribute rules. Rule 4 is an object rule, which works for the *how* attribute of the object *pipeType3* only.

4.3. Rule deployment

This step is concerned with how a context table can be manipulated to restore implicit information. To recover implicit information for an object, we first identify a set of rules applicable to it. This includes all relevant attribute rules and object rules for the object. Each attribute of the object is examined to see if it has implicit information. If the answer is yes, the relevant attribute rule is identified. The identification of an object rule is straightforward as it is linked to the concerned object directly. For an object, if both an attribute rule and an object rule are identified as relevant to an attribute, the object rule overrides the attribute rule when restoring implicit information. For example, for PipeType3 (in Table 1), both Rule 1 and 4 (in Fig. 7) deal with *how* attribute**,** but only Rule 4 is applied when restoring implicit information for *how* of this object.

Once applicable rules have been identified, we generate new objects by applying different combination of the rules. This allows objects with different combination of attributes to be identified. Each derived object retains the existing object attribute relationships of the original object and derives new ones (for attributes having missing values) by applying corresponding rules. For example, for *sewerPipeType1,* there are two attributes that have implicit information, *what* and *location.* Accordingly, two attribute rules are identified: *Rule 1* for *what* attribute and *Rule 2* for *location* attribute. There is no object rule identified for *pipeType1*. By applying different combination of the rules, three new objects are derived from *pipeType1, pipeType1_object1* by applying Rule 1, *pipeType1_object2* by applying Rule 2, and *pipeType1_object3* by applying Rule 1 and 2. All new objects retain existing object attribute relationships of *pipeType1,* and with different relationships derived due to the different rules applied. Depending on the number of rules applicable, each original context object derives different number of new objects. For example, there are 2 applicable rules for *PipeType1, PipeType2* and *PipeType3***.** The combination of these rules generated 3 derived objects for each original object. *PipeType4* has 3 applicable rules and 7 new objects have been derived.

Table 2 lists the many valued context after implicit information has been restored with rules. This many valued context is then fed to Conceptual Scaling component (as shown in Fig. 6) to generate a one valued context table. Table 3 lists the one valued context table after the conceptual scaling of the context in Table 2.

5. Context composition

Context composition takes two formal contexts as input, and generates an integrated GSH. The main components of Context Composition are Context Integration and Hierarchy Generation, as shown in Fig. 8.

The main challenge here is to deal with ambiguous information during context integration, i.e., different terms may be employed to refer to the same attribute, and attributes may be modelled at different levels of granularity. An example here is that one dataset may model a sewerage pipe as either *main* or *lateral* and another may classify it as *trunk main,*

Table 2
A many valued context table after information explication.

	Size	What	How	Location
PipeType1	Main		Pressurised	
pipeType1_object1	Main	Wastewater	Pressurised	
pipeType1_object2	Main		Pressurised	Underground
pipeType1_object3	Main	Wastewater	Pressurised	Underground
pipeType2	Main			Aboveground
pipeType2_object1	Main	Wastewater		Aboveground
pipeType2_object2	Main		Gravity	Aboveground
pipeType2_object3	Main	Wastewater	Gravity	Aboveground
pipeType3	Main	Sludge		
pipeType3_object1	Main	Sludge	Pressurised	
pipeType3_object2	Main	Sludge		Underground
pipeType3_object3	Main	Sludge	Pressurised	Underground
pipeType4	Main			
pipeType4_object1	Main	Wastewater		
pipeType4_object2	Main		Gravity	
pipeType4_object3	Main			Underground
pipeType4_object4	Main	Wastewater	Gravity	
pipeType4_object5	Main	Wastewater		Underground
pipeType4_object6	Main		Gravity	Underground
pipeType4_object7	Main	Wastewater	Gravity	Underground

Please cite this article as: G. Fu, FCA based ontology development for data integration, Information Processing and Management (2016), http://dx.doi.org/10.1016/j.ipm.2016.02.003

Table 3
A one value context table after conceptual scaling of the many valued context in Table 2.

	Main	Wastewater	Sludge	Pressurised	Gravity	Underground	Aboveground
PipeType1	X			X			
pipeType1_object1	X	X		X			
pipeType1_object2	X			X			X
pipeType1_object3	X	X		X			X
pipeType2	X					X	
pipeType2_object1	X	X				X	
pipeType2_object2	X				X	X	
pipeType2_object3	X	X			X	X	
pipeType3	X		X				
pipeType3_object1	X		X	X			
pipeType3_object2	X		X				X
pipeType3_object3	X		X	X			X
pipeType4	X						
pipeType4_object1	X	X					
pipeType4_object2	X				X		
pipeType4_object3	X						X
pipeType4_object4	X	X			X		
pipeType4_object5	X	X					X
pipeType4_object6	X				X		X
pipeType4_object7	X	X			X		X

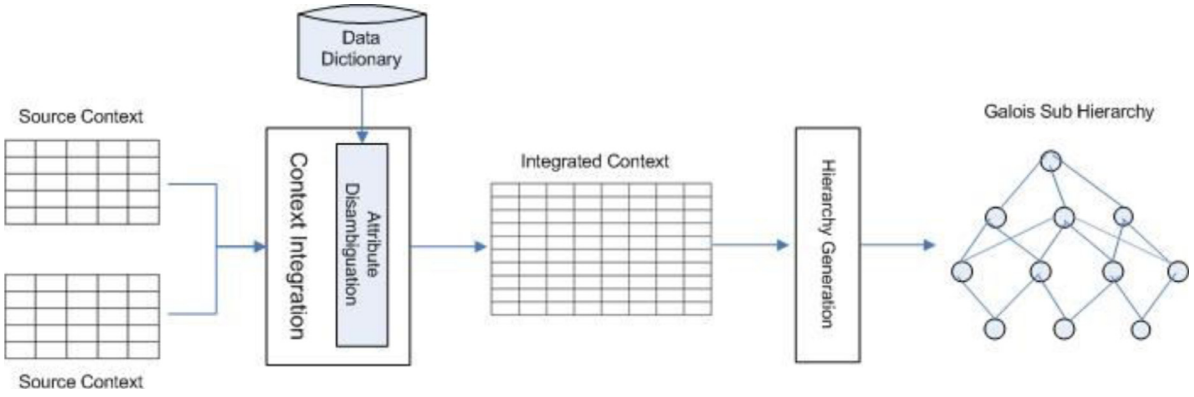

Fig. 8. Context composition.

Table 4
Context K_1.

	Main	Operational	Abandoned	Abandoned intact	Proposed recommission
K1O1	X	X			
K1O2	X				X
K1O3	X		X		
K1O4	X		X	X	

non-trunk main, or *private pipe*. Attribute disambiguation is a process to match attributes from different datasets. In this research we use a pre-defined data dictionary developed in Fu and Cohn (2008a) to disambiguate attributes. The data dictionary maintains a set of terms that describe concepts in a domain, as well as their terminological relationships, e.g. BT/NT (Broader/Narrower Term) etc. Using the data dictionary, we can decide semantic relationships of two attributes. In what follows, we will use the context tables K_1 and K_2 shown in Table 4 and 5 to illustrate the context integration process.

Given two contexts $K_1 := \langle G_1, M_1, I_1 \rangle$ and $K_2 := \langle G_2, M_2, I_2 \rangle$, the integrated context $K := \langle G, M, I \rangle$ is computed by first performing a disjoint union of object sets of two contexts, that is,

$$G = G_1 \cup^* G_2 \tag{1}$$

M and I are assigned M_1 and I_1 from K_1 at this stage, i.e, $M = M_1$ and $I = I_1$. Table 6 shows the context K after the above operations.

The next step identifies the semantic relationship between an attribute in M_2 and an attribute in M. For each attribute $A_i \in M_2$ of K_2, we perform a semantic mapping operation with attributes in K. Based on the mapping identified for A_i, we

Table 5
Context K_2.

	Main	Live	Abandoned destroyed	Proposed	Standby	Abandoned intact
K2O1	X		X			
K2O2	X	X				
K2O3	X			X		
K2O4	X				X	
K2O5	X					X

Table 6
The context table after object union operation.

	Main	Operational	Abandoned	Abandoned intact	Proposed recommission
K1O1	X	X			
K1O2	X				X
K1O3	X		X		
K1O4	X		X	X	
K2O1					
K2O2					
K2O3					
K2O4					
K2O5					

Table 7
Primitive matches.

	attExt	*relExt*
A_i finds an equivalent term A_j in M	Φ	$\{ < O, A_j > \mid if < O, A_i > \in I_2\}$
A_i finds a broader term A_j in M	$\{A_i\}$	$\{ < O, A_i >, <O, Aj > \mid if < O, A_i > \in I_2\}$
A_i finds a narrower term A_j in M	$\{A_i\}$	$\{ < O_m, A_i > \mid if < O_m, A_i > \in I_2\} \cup \{ < O_n, A_i > \mid if < O_n, A_j > \in I\}$
A_i finds no match in M	$\{A_i\}$	$\{ < O, A_i > \mid if < O, A_i > \in I_2\}$

derive the new attributes and relationships to be added to the context table *K*. We use *attExt* to denote the set of attributes to be added to *M*, and use *relExt* to denote the set of relationships to be added to *I*. After each round of mapping, the attribute set *M* and relationship set *I* of *K* are calculated as:

$$M = M \cup attExt \tag{2}$$

$$I = I \cup relExt \tag{3}$$

An attribute could find mappings of different types, including 1 to 1 mapping, or 1 to many mappings, and accordingly different operations for context table manipulation. In what follows, we first describe primitive operations, which deal with 1 to 1 mappings, and we will then discuss how the primitive operations can be composed to deal with 1 to many mappings.

5.1. Primitive operations

For a given attribute $A_i \in M_2$ of K_2, four types of mapping can be identified from *M* of *K*. Table 7 summarises types of mapping, as well as attributes (i.e. *attExt)* and relationships (i.e. *relExt)* to be added to *K* for each mapping type, where A_j denotes the match from *M* of *K*.

I A_i finds an equivalent attribute $A_j \in M$ of *K*. In this case, A_i will be unified with A_j. The context table *K* is expanded with relationships between A_j and objects that have relationships with A_i in K_2. For example for the context K_2 in Table 5, the attribute *live* finds an equivalent attribute *operational* in *K*. They are unified as *operational* in *K*, i.e., $attExt = \Phi$ (no new attribute to be added to *K*). Since there exists a relationship *<K2O2, live>* in K_2, new relationships are established in *K*, i.e., *relExt={<K2O2, operational>}*. Similarly we found an equivalent match *main* in *K* for attribute *main* in K_2. The resultant context table is shown in Table 8, where the newly added relationships are shaded for the purpose of readability.

II A_i finds a match A_j that is more generic to it. In this case, the resulting context *K* is expanded with attribute A_i and relationships between A_i and objects from K_2. New relationships are established in *K* between those objects having attribute A_i and attribute A_j. The theory is that if A_i is a more specific feature of A_j, then any object which has attribute A_i should

Table 8
The context table after an equivalent match.

	Main	Operational	Abandoned	Abandoned intact	Proposed recommission
K1O1	X	X			
K1O2	X				X
K1O3	X		X		
K1O4	**X**		X	X	
K2O1	**X**				
K2O2	**X**	**X**			
K2O3	**X**				
K2O4	**X**				
K2O5	**X**				

Table 9
The context table after a broader match.

	Main	Operational	Abandoned	Abandoned intact	Proposed recommission	Abandoned destroyed
K1O1	X	X				
K1O2	X				X	
K1O3	X		X			
K1O4	X		X	X		
K2O1	X		X			X
K2O2	X	X				
K2O3	X					
K2O4	X					
K2O5	X					

Table 10
The context table after a narrower match.

	Main	Operational	Abandoned	Abandoned intact	Proposed recommission	Abandoned destroyed	Proposed
K1O1	X	X					
K1O2	X				X		X
K1O3	X		X				
K1O4	X		X	X			
K2O1	X		X			X	
K2O2	X	X					
K2O3	X						X
K2O4	X						
K2O5	X						

also have attribute A_j. For example, the closest match for *abandoned destroyed* in K_2 is *abandoned* which is a broader term to it. Then attribute *abandoned destroyed* is added to K, and following two relationships are added to K:

$$< K2O1,\ abandoned\ destroyed >$$

$$< K2O1,\ abandoned >$$

where the first is originated from K_2 due to the existence of *<K2O1, abandoned destroyed>* in K_2. The second is derived due to the fact that *abandoned destroyed* is a more specific feature to *abandoned,* and therefore the existence of *<K2O1, abandoned destroyed>* derives *<K2O1, abandoned>*. The result of these context extensions is highlighted in Table 9.

III A_i finds a match A_j that is more specific to it. In this case the context K is expanded with A_i and existing relationships between A_i and objects from K_2. New binary relationships are established in K between those objects having relationships with A_j (originally from K1) and attribute A_i (which is originally from K_2). The theory is that if A_i is a more generic feature of A_j, then any object which has attribute A_j should also have attribute A_i. For example, if the closest match for the attribute *proposed* in K_2 is *proposed recommission* in K which is a narrower term to it, then attribute *proposed* is added to K. The following two relationships are added to K,

$$< K2O3,\ proposed >$$

$$< K1O2,\ proposed >$$

where the first is originated from K_2 due to the existence of *<K2O3,proposed>* in K_2. The second is established due to the existence of *<K1O2,proposed recommission>* in K as well as the fact that *proposed* is a more generic feature to *proposed recommission.* The result of this mapping operation is shown in Table 10.

Please cite this article as: G. Fu, FCA based ontology development for data integration, Information Processing and Management (2016), http://dx.doi.org/10.1016/j.ipm.2016.02.003

Table 11
The context table after a non-match.

	Main	Operational	Abandoned	Abandoned intact	Proposed recommission	Abandoned destroyed	Proposed	Standby
K1O1	X	X						
K1O2	X				X		X	
K1O3	X		X					
K1O4	X		X	X				
K2O1	X		X			X		
K2O2	X	X						
K2O3	X						X	
K2O4	X							X
K2O5	X							

Table 12
The context table after a composite match (the integrated context table).

	Main	Operational	Abandoned	Abandoned intact	Proposed recommission	Abandoned destroyed	Proposed	Standby
K1O1	X	X						
K1O2	X				X		X	
K1O3	X		X					
K1O4	X		X	X				
K2O1	X		X			X		
K2O2	X	X						
K2O3	X						X	
K2O4	X							X
K2O5	X		**X**	**X**				

IV A_i finds no match in K. In this case the context K is simply expanded with A_i and existing relationships between A_i and objects originating from K_2. For example there is no semantic match in K for the attribute *standby* of K_2. In this case, K is extended with $attExt = \{standby\}$ and $relExt = \{<K2O4, standby>\}$, as shown in Table 11.

5.2. Composite operations

In many situations, an attribute may have multiple matches and each match is of different type, e.g. having both an equivalent and a broader match at same time. The primitive operations discussed above can be composed to deal with these complex cases. For an attribute $A \in M_2$ of K_2, if a set of matches $\{A_1, A_2, ..., A_n\}$ are identified from M of K, the context K is extended as follows:

$$M = M \bigcup_{j=1}^{n} attExt_{A_j} \tag{4}$$

$$I = I \bigcup_{j=1}^{n} relExt_{A_j} \tag{5}$$

where $attExt_{A_j}$ and $relExt_{A_j}$ respectively denote the attribute and relationship sets that are derived when A is matched to A_j with the primitive operations discussed in Section 5.1.

For example, for the attribute *abandoned intact*, two matches are found from K, the equivalent match *abandoned intact* and the generic match *abandoned.* For equivalent match *abandoned intact*, the following are generated:

$$attExt = \Phi$$

$$relExt = \{< K2O5,\ abandoned\ intact >\}$$

For the generic match *abandoned,* the following are generated by

$$attExt = \{abandoned\ intact\}$$

$$relExt = \{< K2O5,\ abandoned\ intact >, < K2O5,\ abandoned >\}$$

Adding these into K results in the formal context shown in Table 12, which is also final integrated context table. The GSH constructed from this integrated context is illustrated in Fig. 9.

Please cite this article as: G. Fu, FCA based ontology development for data integration, Information Processing and Management (2016), http://dx.doi.org/10.1016/j.ipm.2016.02.003

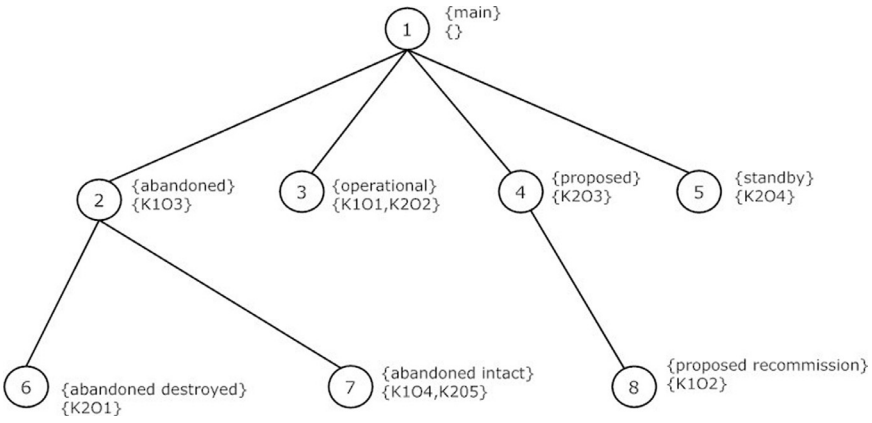

Fig. 9. The GSH derived from the context in Table 12.

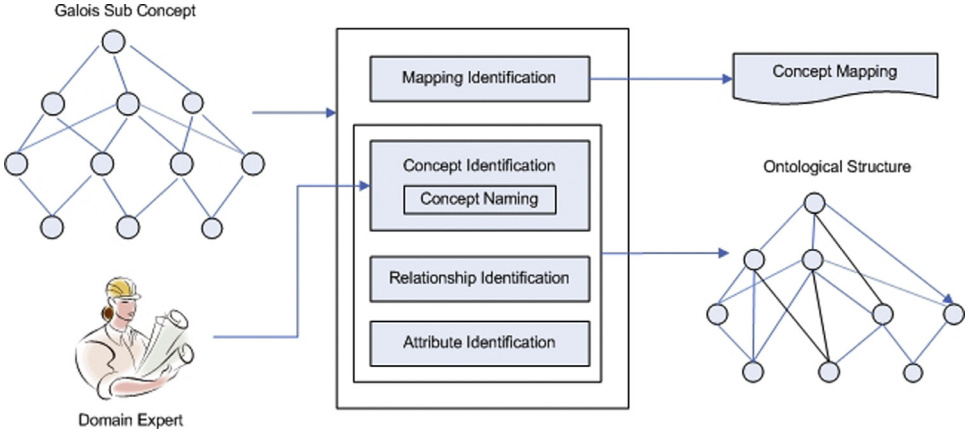

Fig. 10. Ontology derivation.

6. Ontology derivation

Ontology derivation component of our framework takes the GSH generated in Section 5 and generates an ontological structure[1]. Fig. 10 shows the components of ontology derivation. The GSH is exploited to derive several types of information, including ontological concepts, subsumption relationships between concepts, and attributes of concepts. The information identified forms an ontological structure from which a full ontology can be developed. The mapping between concepts of different datasets can also be identified from the GSH.

6.1. Mapping identification

This subcomponent derives mappings between concepts of two datasets. Given a formal concept in a GSH, if its extent contains more than one objects (e.g. the extent of node 3 is {K1O1, K2O2} in Fig. 9), then it indicates a potential mapping between these source concepts. The validation of domain engineers is requested at the evaluation stage to judge whether a mapping identified is correct. If the answer is negative, features need to be identified to differentiate one concept from another. This often involves the identification of new attributes or relationships of concerned concepts. The existence of incorrect matches triggers the need to iterate context composition or integration operations.

6.2. Concept/relationship/attribute identification

As we employ a GSH in the research, intermediate, abstract concepts are reduced in the context integration step and the resulting hierarchy consists only of object concepts and attribute concepts. Object concepts have to be kept in the resultant ontological hierarchy as they correspond to the initial concepts (either explicit or implicit) of datasets and therefore need

[1] The term *ontological structure* is used here to mean that the derived conceptual structure only contains limited data semantics, i.e. only concepts, attributes and *is-a* relationships are identified. Further development is still required to capture other data semantics to generate a full ontology.

Please cite this article as: G. Fu, FCA based ontology development for data integration, Information Processing and Management (2016), http://dx.doi.org/10.1016/j.ipm.2016.02.003

Table 13
Source datasets.

Dataset	No of concepts	No of attributes
*D*1	54	18
*D*2	47	21
*D*3	43	22
*D*4	66	23

to remain in ontological structure to respect the initial class specification of the datasets. For an attribute concept, the assistance of domain engineers is required to decide whether it should be kept or discarded by taking into account its significance or interest to the application. When an attribute concept is discarded in a GSH, all elements in its intent are passed on to its sub concepts, and *super-/sub-concept* relationships are established between its super-concepts and sub-concepts.

After a decision has been made on which concepts are to be kept in the resultant ontology, the rules for identifying relationships and attributes of a concept are straightforward:

- All elements in the intent of a formal concept are declared as attributes of the ontological concept.
- Sub/super relations between two formal concepts are identified as *is-a* relationships between the corresponding ontological concepts.

7. Empirical evaluation

An evaluation of the proposed techniques has been performed on several industrial datasets. We first describe the experimental setup and the ontology similarity measures employed in the evaluation. We then report on the evaluation results.

7.1. Experiment setup and ontology similarity measures

Datasets we used for performing our experiments were sourced from four UK water companies. These datasets essentially encode same types of information, including various water pipes, metering and treatment facilitates for transporting freshwater/wastewater for customers across the UK. However each organisation records its information with little thought towards interoperability with others. This results in data heterogeneities. Due to data confidentiality agreement we have with our industrial partners, we cannot publish these datasets. Nevertheless we have list in Table 13 the statistics on the datasets.

The mapping and integration was carried out in a semi-automated manner, where data acquisition and attribute disambiguation were conducted manually, the open source tool Galicia (Valtchev et al., 2003) was employed for context manipulation and GSH generation, and all other processes such as information explication and conceptual scaling were completed with Java and SQL codes. The evaluation was performed in three phases.

- Phase I experiments constructed *local* ontologies for each dataset involved. Pairwise comparison was conducted to measure the similarity of these local ontologies, and the results were then served as benchmarks for the subsequent evaluation.
- Phase II experiments studied how implicit information impacts on ontology interoperability, and demonstrated how information explication can help with ontology alignment.
- Phase III compared an ontology developed in this research with a handcrafted ontology developed with traditional knowledge engineering approach. The performance of two ontologies was evaluated by studying how best the two ontologies fit and respect the knowledge structures of datasets to be integrated.

Evaluation was performed at 2 levels: lexical level and taxonomic level. Lexical level evaluation reflects how well the lexical terms of a source ontology cover those of a target ontology. Taxonomic level evaluation examines how well the conceptual hierarchy of a source ontology resembles that of a target ontology. We employ the ontology measures proposed in Dellschaft and Staab (2006) and Maedche and Staab (2002) in our experiments. Lexical precision and recall of a source ontology O_S against a target Ontology O_T are computed as:

$$LP(O_S, O_T) = \frac{|C_S \cap C_T|}{|C_S|} \tag{6}$$

$$LR(O_S, O_T) = \frac{|C_S \cap C_T|}{|C_T|} \tag{7}$$

where C_S (or C_T) is the set of terms describing concepts in O_s(or O_T).

Lexical F-measure, *LF*, is used for balancing the precision and recall values, and is calculated as harmonic mean of *LP* and *LR*.

$$LF = \frac{2 \cdot LP(O_S, O_T) \cdot LR(O_S, O_T)}{LP(O_S, O_T) + LR(O_S, O_T)} \tag{8}$$

Please cite this article as: G. Fu, FCA based ontology development for data integration, Information Processing and Management (2016), http://dx.doi.org/10.1016/j.ipm.2016.02.003

Taxonomic level measures are divided into local and global measures. Local measures compare the similarity of hierarchical positions of two concepts in the source and the target ontologies. For local taxonomic precision, the similarity of two concepts is computed based on the *common semantic cotopies* from the concept hierarchies. The *common semantic cotopies* includes all the common super- and sub-concepts of a concept pair. Given such a semantic cotopy *ce*, the local taxonomic precision *tp* and recall *tr* of two concepts $c_1 \in C_S$ and $c_2 \in C_T$ is defined as

$$tp(c_1, c_2, O_S, O_T) = \frac{|ce(c_1, O_S) \cap ce(c_2, O_T)}{|ce(c_1, O_S)|} \tag{9}$$

$$tr(c_1, c_2, O_S, O_T) = \frac{|ce(c_1, O_S) \cap ce(c_2, O_T)}{|ce(c_2, O_T)|} \tag{10}$$

Since $tp(c_2, c_1, O_T, O_S) = \frac{|ce(c_1, O_S) \cap ce(c_2, O_T)}{|ce(c_2, O_T)|}$, we have

$$tr(c_1, c_2, O_S, O_T) = tp(c_2, c_1, O_T, O_S) \tag{11}$$

Global taxonomic precision and recall are defined by summing up local taxonomic precision and recall of common concepts in two ontologies.

$$TP(O_S, O_T) = \frac{1}{|C_S \cap C_T|} \sum_{c \in (C_S \cap C_T)} tp(c, c, O_S, O_T) \tag{12}$$

$$TR(O_S, O_T) = \frac{1}{|C_S \cap C_T|} \sum_{c \in (C_S \cap C_T)} tr(c, c, O_S, O_T) \tag{13}$$

Since $TP(O_T, O_S) = \frac{1}{|C_S \cap C_T|} \sum_{c \in (C_S \cap C_T)} tp(c, c, O_T, O_S)$ and $tp(c, c, O_T, O_S) = tr(c, c, O_S, O_T)$ due to Eq. (11), we have

$$TR(O_S, O_T) = TP(O_T, O_S) \tag{14}$$

Taxonomic F-measure *TF* is used to balance *TP* and *TR* to generate a combined taxonomic measure.

$$TF(O_S, O_T) = \frac{2 * TP(O_S, O_T) * TR(O_S, O_T)}{TP(O_S, O_T) + TR(O_S, O_T)} \tag{15}$$

A combined measure *GF*, which balances the lexical and taxonomic measures, is used to give a summarising overview of the similarity of O_S against O_T, and is computed as the harmonic mean of *LF* and *TF*:

$$GF(O_S, O_T) = \frac{2 * LF(O_S, O_T) * TF(O_S, O_T)}{LF(O_S, O_T) + TF(O_S, O_T)} \tag{16}$$

7.2. Experimental results

7.2.1. Phase I

The concepts and attributes from four datasets have been identified and used to generate context tables. The context tables were then fed to Galicia to derive GSHs. An ontology was generated from a GSH by discarding all attribute objects and keeping the object concepts. Four ontologies were generated, each for a dataset (i.e. an ontology *Oi* is generated for a dataset *Di* where $i = 1, 2, 3$ and 4).

Matrices described in Section 7.1 were used to measure the similarity of these ontologies. We observed that the four water companies differ greatly from each other on what business objects they record in their systems, which leads to ontologies that are incompatible to each other both lexically and taxonomically. These local ontologies only agreed with each other to a small extent: only a relatively small percentage of terms in one ontology were also found in another ontology. This was measured with lexical precision LP (Table 14). Ontology *O2* is the one that has the least common terms with other ontologies. Manual inspection of these ontologies found that this lexical disagreement was mainly due to the different aspects of the domain that an organisation chose to encode in its data management systems, and this resulted in different ontology concepts. The poor performance of *O2* ontology was due to the granularity issues—it encoded concepts at a finer level than other ontologies, which resulted in lexical mismatches with other ontologies.

Table 14
Baseline *LP*.

	O1	*O2*	*O3*	*O4*
O1	–	9.26%	20.37%	25.93%
O2	10.38%	–	8.51%	17.02%
O3	25.58%	9.30%	–	27.91%
O4	21.21%	12.12%	18.19%	–

Please cite this article as: G. Fu, FCA based ontology development for data integration, Information Processing and Management (2016), http://dx.doi.org/10.1016/j.ipm.2016.02.003

Table 15
Baseline *TP*.

	O1	*O2*	*O3*	*O4*
O1	–	10.00%	40.90%	34.52%
O2	11.83%	–	11.07%	26.82%
O3	36.60%	12.60%	–	30.75%
O4	33.29%	24.58%	30.65%	–

Table 16
Rule sets.

	Total no of rules	No of attribute rules	No of object rules
O1	15	6	9
O2	12	5	6
O3	11	6	5
O4	15	6	9

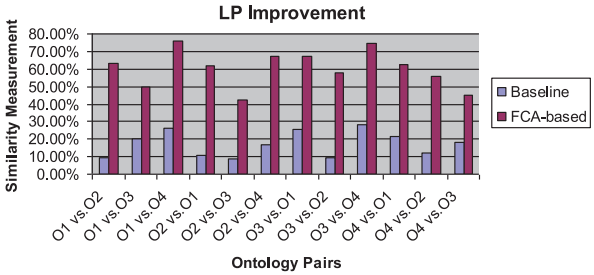

Fig. 11. Lexical precision.

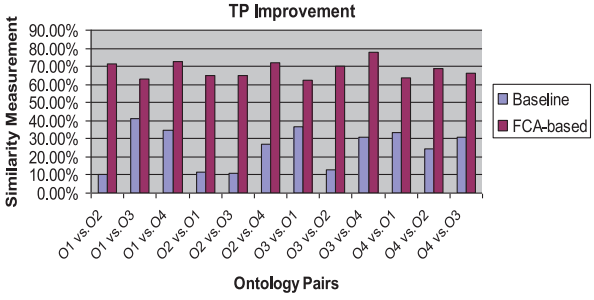

Fig. 12. Taxonomic precision.

The taxonomic level similarity of these ontologies was slightly better but scores were still quite low, as shown in Table 15. The presence of different concepts in the hierarchies of these ontologies led to disappointing results. Again, ontology *O2* performed the worst—it has a much lower taxonomic precision when compared to the other ontologies. Examination revealed that the granularity mismatch was again the main cause for this. As *O2* ontology encoded business objects at a finer granularity than others, it had a very different hierarchy to those of other ontologies.

7.2.2. Phase II

Experiments were firstly performed to restore implicit information for ontologies generated in Phase I, the resultant ontologies were then compared with each other using same measures. To do this, rules for restoring implicit information were acquired for each dataset with the help of domain engineers. Table 16 shows the statistics on these rule sets.

The rules were deployed to formal contexts generated in the Phase I experiments to restore implicit information as well as derive new feature types. The resultant contexts were used to generate ontologies in the same way as did in Phase I experiments. The similarity measures were calculated for these ontologies, and the results were compared with the ones we obtained in Phase I experiment, which are shown in Figs. 11 and 12. Comparing with the baseline similarity scores, we can see a substantial improvement in the similarity of these ontologies, both at the lexical level and at the taxonomic level. The average lexical precision increased to around 60% which was below 20% in the Phase I study (Fig. 11). This was mainly due to the increase of the common feature types which were restored in information explication process.

Please cite this article as: G. Fu, FCA based ontology development for data integration, Information Processing and Management (2016), http://dx.doi.org/10.1016/j.ipm.2016.02.003

Table 17
Attribute disambiguation.

Target	Source	Exact match	BT match	NT match	Non-match	Multi-match
O1	*O2*	12	0	2	5	2
O1+O2	*O3*	15	1	1	4	1
O1+O2+O3	*O4*	16	2	2	2	1

Table 18
Performance of FCA ontology and KE ontology.

	FCA ontology			*KE ontology*		
	LP	*TP*	*GF*	*LP*	*TP*	*GF*
O1	64.11%	67.07%	83.81%	62.88%	53.29%	44.39%
O2	64.92%	74.61%	88.02%	53.61%	51.81%	39.67%
O3	47.18%	64.99%	82.93%	52.58%	49.67%	46.25%
O4	72.39%	75.54%	86.38%	71.13%	52.41%	42.17%

Taxonomic precision was improved similarly: from 20% to around 60% by average (Fig. 12). This improvement was mainly due to the resulting ontologies bearing a similar level of detail in their hierarchies once they were enriched with derived objects generated with rules. A concept in one ontology had an increased number of common super- and sub-concepts with its matching concept in another ontology. This resulted in improved local taxonomic similarity and therefore improved global taxonomic similarity. This led to the conclusion that implicit information impacts greatly on the similarity of the local ontologies, and similarity of these ontologies can be improved significantly if we can have implicit information restored.

7.2.3. Phase III

The four local ontologies, which had implicit information restored in Phase II experiments, were then integrated to build a global ontology. This was achieved by first performing the context integration as described in Section 5. The contexts of *O1* and *O2* were integrated first, and resultant context was then integrated with *O3* context and so on, as shown in Table 17. The main activity performed here was attribute disambiguation. Table 17 shows the types of attribute matches found during various stages of the integration process. For example, for the 21 attributes of the *O2* context, 12 found an exact match from the *O1* context, and 2 found narrower matches, 5 did not find any match, and 2 found multiple matches. After attribute disambiguation, the integrated context was used to generate a GSH, from which an integrated ontology was derived. The total number of concepts in the integrated ontology was 248 and the depth of hierarchy was 6.

To evaluate the quality of this integrated ontology (*FCA ontology* for short), we compared it against a handcrafted ontology that was developed with a traditional knowledge engineering approach as described in Fu and Cohn (2008a) (*KE ontology* for short). Both *FCA ontology* and *KE ontology* had the same local ontologies (i.e. *O1, O2, O3* and *O4*) as major inputs (and therefore comparison made in this research are unbiased), but they differ from each other on how ontological hierarchies were built and how implicit/unarticulated information was recovered. The hierarchy of FCA ontology was generated automatically with FCA tool Glacia basing on the attribute definition of objects, and the hierarchy of KE ontology was generated manually basing on the domain knowledge from domain experts. *FCA ontology* achieved information explication via the domain rules as discussed in Section 4. *KE ontology* did this through a manual semantic enrichment process. Extra data semantics of *KE ontology* were manually derived from both system design documents and domain engineers. The resultant *KE ontology* consists of 216 concepts which was organised in 5 hierarchical levels.

We evaluate the two ontologies in the similar fashion as done in (Brewster, Alani, Dasmahapatra, & Wilks, 2004). We consider that an ontology is of good quality when it conforms to and has a good coverage of knowledge structures of datasets to be integrated. This was performed by comparing *FCA ontology* and *KE ontology* against local ontologies *O1, O2, O3* and *O4* as developed in Phase II experiment. Table 18 summarises the results. Both ontologies had similar scores for the lexical precision LP when compared against these ontologies. This can be largely explained by that both *FCA ontology* and *KE ontology* had these local ontologies as input, i.e., concepts in these ontologies were major lexical sources of both ontologies. *FCA ontology* outperformed *KE ontology* on its similarity to the local ontologies at the taxonomic level. This is because *FCA ontology* was generated systematically based on attribute definitions of input feature types (of the local ontologies), and sub- and super-concept relationships between concepts were identified in the same fashion as the local ontologies. This led to the improved taxonomic precision of the *FCA ontology*. However the ontological hierarchy generated with KE method is rather subjective, i.e. depending upon human judgement on what intermediate concept to add, and when a sub-/super-concept relationship should be established. The hierarchy tends to be distorted with missing sub- and super-concept links when the number of concepts increases. *FCA ontology* also outperformed *KE ontology* on the overall similarity measure *GF*. This leads to the conclusions that *FCA ontology* fits and respects the local ontologies better and therefore better serves the integration purpose in this case.

Please cite this article as: G. Fu, FCA based ontology development for data integration, Information Processing and Management (2016), http://dx.doi.org/10.1016/j.ipm.2016.02.003

8. Conclusions and future work

The availability of vast quantities of data presents organisations with both opportunities and challenges. Data integration techniques offer a promising way for addressing the issue of data heterogeneities and promoting data sharing and interoperability across organisations. In this paper we present a formal and semi-automated method for ontology development, with the aim to reconcile heterogeneous data and support data integration. The research extends classical FCA theory to address the issues of implicit and ambiguous information, which, we consider, are important but have not been sufficiently investigated by previous studies. The research enables considerable ontology engineering activities automated, including concept derivation and hierarchy generation. In contrast to studies that draw upon either small or simplified datasets, we evaluate the proposed techniques on non-trivial industrial datasets. Our experimental results demonstrate the techniques described in this paper can help curate and fuse data from disperse sources, and support the development of ontologies that better *fits* and *respects* the underlying knowledge structure of domain. There are a number of works which we plan to undertake in the future, including developing techniques to deal with incomplete information in data integration, and validating the proposed techniques on datasets in other application domains.

Acknowledgments

This research is a follow on work of UK EPSRC grant (EP/C014707/1). Gaihua Fu is currently funded by the EPSRC grants (EP/I035781/1) and EP/K012398/1.

References

Bahga, A., & Madisetti, V. K. (2015). Healthcare data integration and informatics in the cloud. *Computer, 48*(2), 50–57.
Bai, X., & Zhou, X. Z. (2011). Development of ontology-based information system using formal concept analysis and association rules. *Advances in Computer Science, Intelligent System and Environment, 106*, 121–126.
Bakhtouchi, A., Bellatreche, L., & Ait-Ameur, Y. (2011). Ontologies and functional dependencies for data integration and reconciliation. *Advances in Conceptual Modeling: Recent Developments and New Directions, 6999*, 98–107.
Beck, A., Boukhelifa, N., Fu, G., Hickinbotham, S., Parker, J., Bennett, B., et al. (2013). Utility data integration and knowledge representation in the UK: The VISTA project. *GeoHydroinformatics - Integrating GIS and Water Engineering*. United Kingdom: Chapman & Hall/CRC Press.
Bian, J., Zhang, H., & Peng, X. G. (2011). The research and implementation of heterogeneous data integration under ontology mapping mechanism. *Web Information Systems and Mining, 6988*, 87–94.
Brewster, C., Alani, H., Dasmahapatra, S., & Wilks, Y. (2004). Data-driven ontology evaluation. In *Proceedings of the Language Resources and Evaluation Conference* (pp. 164–168).
Brockmans, S., Colomb, R. M., Haase, P., Kendall, E. F., Wallace, E. K., Welty, C., et al. (2006). A model driven approach for building OWL DL and OWL full ontologies. In *Proceedings of the International Semantic Web Conference (ISWC): 2006* (pp. 187–200).
Chen, R. C., Bau, C. T., & Yeh, C. J. (2011). Merging domain ontologies based on the WordNet system and Fuzzy Formal Concept Analysis techniques. *Applied Soft Computing, 11*(2), 1908–1923.
Dellschaft, K., & Staab, S. (2006). On how to perform a gold standard based evaluation of ontology learning. In *Proceedings of the International Semantic Web Conference (ISWC): 2006* (pp. 228–241).
Do, H., & Rahm, E. (2002). COMA - a system for flexible combination of schema matching approaches. In *Proceedings of the 28th International Conference on Very Large Data Bases, August 20-23, 2002* (pp. 610–621). Hong Kong, China.
Doan, A., Domingos, P., & Halevy, A. Y. (2001). Reconciling schemas of disparate data sources: A machine-learning approach. In *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data, May 21-24, 2001* (pp. 509–520). Santa Barbara, CA, USA.
Doan, A., Halevy, A., & Ives, Z. G. (2012). *Principles of Data Integration*. Waltham, MA: Morgan Kaufmann.
Doan, A., Madhavan, J., Domingos, P., & Halevy, A. Y. (2003). Ontology matching: A machine learning approach. In Staab Steffen, et al. (Eds.), *Handbook on Ontologies* (pp. 397–416). Berlin Heidelberg: Springer.
Doan, A., Noy, N. F., & Halevy, A. Y. (2004). Introduction to the special issue on semantic integration. *Sigmod Record, 33*(4), 11–13.
Duckham, M., & Worboys, M. (2005). An algebraic approach to automated information fusion. *International Journal of Geographic Information Systems, 19*(5), 537–557.
Duong, T. H., & Jo, G. S. (2012). Enhancing performance and accuracy of ontology integration by propagating priorly matchable concepts. *Neurocomputing, 88*, 3–12.
Duong, T. H., Truong, H. B., & Nguyen, N. T. (2012). Local neighbor enrichment for ontology integration. *Intelligent Information and Database Systems, 7196*, 156–166.
Formica, A. (2006). Ontology-based concept similarity in formal concept analysis. *Information Sciences, 176*(18), 2624–2641.
Fu, G., & Cohn, A. G. (2008a). Semantic integration for mapping the underworld. In *Geoinformatics 2008 and Joint Conference on GIS and Built Environment: Geo-Simulation and Virtual GIS Environments, June 28-29, 2008*. Guangzhou, China. Doi: 7143/714327-714327-714329.
Fu, G., & Cohn, A. G. (2008b). Utility ontology development with formal concept analysis. In *5th International Conference on Formal Ontology in Information Systems* (pp. 297–310).
Ganter, B., Stumme, G., & Wille, R. (2005). *Formal Concept Analysis: Foundations and Applications*. Berlin, Heidelberg: Springer.
Ganter, B., & Wille, R. (1999). *Formal Concept Analysis: Mathematical Foundations*. Berlin, New York: Springer.
Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition, 5*(2), 199–220.
He, L. J., & Wang, Q. T. (2011). Construction of ontology information system based on formal concept analysis. *Advances in Computer Science, Intelligent System and Environment, 104*, 83–88.
Huang, S. L., Lin, S. C., & Chan, Y. C. (2012). Investigating effectiveness and user acceptance of semantic social tagging for knowledge sharing. *Information Processing & Management, 48*(4), 599–617.
Jiang, Y. C., Zhang, X. P., Tang, Y., & Nie, R. H. (2015). Feature-based approaches to semantic similarity assessment of concepts using Wikipedia. *Information Processing & Management, 51*(3), 215–234.
Kalfoglou, Y., & Schorlemmer, M. (2003). Ontology mapping: The state of the art. *The Knowledge Engineering Review, 19*(1), 1–31.
Lenzerini, M. (2002). Data integration: A theoretical perspective. In *Proceedings of the 21st ACM Sigmod-Sigact-Sigart Symposium on Principles of Database Systems* (pp. 233–246).
Liu, J., & Zhang, X. X. (2014). Data integration in fuzzy XML documents. *Information Sciences, 280*, 82–97.
Madhavan, J., Bernstein, P. A., & Rahm, E. (2001). Generic schema matching with cupid. In *Proceedings of the 27th International Conference on Very Large Data Bases* (pp. 49–58).

Please cite this article as: G. Fu, FCA based ontology development for data integration, Information Processing and Management (2016), http://dx.doi.org/10.1016/j.ipm.2016.02.003

Madhavan, J., & Halevy, A. (2003). Composing mappings among data sources. In *Proceedings of the 29th International Conference on Very Large Data Bases* (pp. 572–583).
Maedche, A., & Staab, S. (2002). Measuring similarity between ontologies. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management.: 2473:* (pp. 251–263).
Mate, S., Kopcke, F., Toddenroth, D., Martin, M., Prokosch, H. U., Burkle, T., et al. (2015). Ontology-based data integration between clinical and research systems. *Plos One, 10*(1), E0122172.
Nanda, J., Simpson, T. W., Kumara, S. R. T., & Shooter, S. B. (2006). A methodology for product family ontology development using formal concept analysis and Web ontology language. *Journal of Computing and Information Science in Engineering, 6*(2), 103–113.
Neumann, F., Ho, C., Tian, X., Haas, L., & Meggido, N. (2002). Attribute classification using feature analysis. In *Proceedings of the 18th International Conference on Data Engineering* (p. 271).
Nguyen, N. T. (2006). Conflicts of ontologies - classification and consensus-based methods for resolving. In *Knowledge-Based Intelligent Information and Engineering Systems, Pt 2, Proceedings: 4252* (pp. 267–274).
Nguyen, N. T. (2007). A method for ontology conflict resolution and integration on relation level. *Cybernetics and Systems, 38*(8), 781–797.
Noy, N. F. (2004). Semantic Integration: A survey of ontology-based approaches. *Sigmod Record, 33*(4), 65–70.
Noy, N. F., & Musen, M. A. (2000). PROMPT: Algorithm and tool for automated ontology merging and alignment. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence* (pp. 450–455).
Pedersen, T. B., Pedersen, D., & Riis, K. (2013). On-demand multidimensional data integration: Toward a semantic foundation for cloud intelligence. *Journal of Supercomputing, 65*(1), 217–257.
Pinto, H. S., & Martins, J. P. (2004). Ontologies: How can they be built? *Knowledge and Information Systems, 6*(4), 441–464.
Rahm, E., & Bernstein, P. A. (2001). A survey of approaches to automatic schema matching. *The VLDB Journal, 10*(4), 334–350.
Rodriguez, M., & Egenhofer, M. (2003). Determining semantic similarity among entity classes from different ontologies. *IEEE Transactions on Knowledge and Data Engineering, 15*(2), 442–456.
Rouane, M., Valtchev, P., Sahraoui, H., & Huchard, M. (2004). Merging conceptual hierarchies using concept lattices. In *Proceedings of the 3rd Workshop on Managing SPEcialization/Generalization Hierarchies, June 15, 2004* (pp. 51–58). Oslo, Norway.
Spohr, D., Hollink, L., & Cimiano, P. (2011). A machine learning approach to multilingual and cross-lingual ontology matching. In *Proceedings of the 10th International Conference on the Semantic Web (ISWC) 2011: 7031:* (pp. 665–680).
Stumme, G., & Maedche, A. (2001). FCA-MERGE: Bottom-up merging of ontologies. In *International Joint Conference On AI, August 4-10, 2001* (pp. 225–234). Seattle, Washington, USA.
Sure, Y., Tempich, C., & Vrandecic, D. (2006). Ontology engineering methodologies. *Semantic Web Technologies: Trends and Research in Ontology-based Systems* (pp. 171–190). Wiley.
Truong, H. B., & Nguyen, N. T. (2012). A multi-attribute and multi-valued model for fuzzy ontology integration on instance level. *Intelligent Information and Database Systems, 7196*, 187–197.
Uschold, M., & Grüninger, M. (2004). Ontologies and semantics for seamless connectivity. *Sigmod Record, 33*(4), 58–64.
Valtchev, P., Grosser, D., Roume, C., & Hacene, M. R. (2003). Galicia: an open platform for lattices. In *Proceedings of the 11th International Conference on Conceptual Structures, July 21-25, 2003* (pp. 241–254). Dresden, Germany.
Wille, R., & Reidel, I. R. (1982). Restructuring lattice theory: An approach based on hierarchies of concepts. *Ordered Sets* (pp. 445–470). Dordrecht, Boston.
Xia, H. (2013). Semantic web ontology integration based on formal concept analysis. *Mechatronics, Robotics and Automation, 373-375*, 1714–1718.
Xie, J., Liu, F., & Guan, S. U. (2011). Tree-structure based ontology integration. *Journal of Information Science, 37*(6), 594–613.
Yang, S. (2011). Efficient ontology integration model for better inference in context aware computing. *Computational Materials Science, 268-270*, 841–846.
Yu, T., Chen, H. J., Mi, J. H., Gu, P. Q., Wu, T., & Pan, J. Z. (2012). DartWiki: A semantic wiki for ontology-based knowledge integration in the biomedical domain. *Current Bioinformatics, 7*(3), 278–288.
Zhao, Y., Wang, X., & Halang, W. (2006). Ontology mapping based on rough formal concept analysis. In *Advanced International ConfeSence on Telecommunications and International Conference on Internet and Web Applications and Services (AICT/ICIW 2006)* 19-25 February 2006. French Caribbean: Guadeloupe.

Please cite this article as: G. Fu, FCA based ontology development for data integration, Information Processing and Management (2016), http://dx.doi.org/10.1016/j.ipm.2016.02.003